Ubuntu GNU/Linux

Eine Einführung

M. Sabath

22. September 2006

Inhaltsverzeichnis

1 Grundlagen

1.1 Begriffe....3
1.2 Ubuntu....4

2 Ubuntu, Installation und Support im WWW

2.1 Gründe umzusteigen....5
2.2 Hardware: Linux ist nicht gleich Windows....5
2.3 Bezugsquellen Ubuntu....6
2.4 Die Ubuntu Community....6
2.5 Installation von Ubuntu....8
2.6 Bezeichnung von Festplatten ***....9
2.7 Ubuntu ist installiert....10

3 Es wäre schön root zu sein

3.1 Der normale Nutzer unter Linux....11
3.2 Es gibt kein root....11
3.3 Root Rechte erlangen....12

4 Software installieren

4.1 Vorbereitung: /etc/apt/sources.list....13
4.2 Abhängigkeiten ***....14
4.3 Unter Gnome....14
4.4 Im Terminal / in der Konsole....14
4.5 Starten von Programmen ***....15
4.6 Ubuntu und Multimedia....16
4.7 Microsoft Schriftarten....17
4.8 Panels erweitern....17

5 Zugriffsrechte unter Linux *

5.1 Grundlagen....18
5.2 Anzeigen der Zugriffsrechte....18
5.3 Zugriffsrechte mit Nautilus ändern....20
5.4 Zugriffsrechte mit chmod ändern....20

6 Linux konfigurieren

6.1 Textdateien ***....22
6.2 Systemweite Konfiguration....22
6.3 Benutzerspezifische Konfiguration....23
6.4 Datensicherung....23
6.5 Fehlersuche....23
6.6 Nichts geht mehr....24

7 Editoren und einige Tipps

7.1 Editoren....25
7.2 Die Tabulatortaste ***....27
7.3 Die Pfeiltasten ***....27
7.4 Mittlere Maustaste ***....28
7.5 Abbruch von Programmen ***....28
7.6 Den X-Server neu starten ***....28
7.7 Linux reagiert nicht mehr ***....28
7.8 Linux ausschalten ***....28
7.9 Verzeichnisangaben ***....29
7.10 Pipes: Ausgaben an einen Befehl weiterleiten ***...29
7.11 Ausgaben in eine Datei umleiten / an eine Datei anhängen ***....29

8 Häufige Befehle unter Linux

8.1 Hilfe vom System....30
8.2 Informationen Rund um das System....32
8.3 Umgang mit Dateien/Verzeichnissen....35
8.4 Verwaltung des Systems....41
8.5 Netzwerk....43
8.6 Drucken....44
8.7 Suchen....44

9 Anhang

9.1 root, es gibt ihn doch....47

1 Grundlagen

1.1 Begriffe

Linux

Der Finne Linus Torvald entwickelte 1991 (* 28.12.1969) auf der Basis des Unix ähnlichen Betriebssystems *Minix* ein Programm um mit seinem Computer auf den Server der Universität zugreifen zu können. Aus diesem Programm entwickelte er ein Betriebssystem welches er *Freax* nannte (von dem Wort *freak* und dem x als Hinweis auf Unix). Dieses Betriebssystem stellte er von Beginn an der Allgemeinheit auf einem FTP Server zum Download zur Verfügung. Der Server Administrator speicherte das Programm jedoch nicht in einem Ordner *Freax*, sondern in einem Ordner *Linux* ab. Anfangs war Linus Torvald gegen diesen Namen, sah jedoch schnell ein, dass dies der bessere Name war.

GNU/Linux***

Ab 1993 veröffentlichte Linus Torvald sein Betriebssystem Linux unter der GPL[1] (GNU Gerneral Public License). Erst dadurch wurde die schnelle Verbreitung von Linux möglich. Bei Linux handelt es sich eigentlich nur um den Kern des Betriebssystems den so genannten Kernel[2]. Nur mit einem Kernel alleine ist es nicht getan. An dieser Stelle kommt das GNU[3] Projekt zum Tragen. Um mit einem Computer arbeiten zu können, wird Software benötigt. Genau diese freie Software steht durch das GNU Projekt zur Verfügung. Jeder kann sich somit seine eigene Distribution aus dem Linux Kernel und Software aus dem GNU Projekt „zusammen bauen“.

Distributionen***

Unter einer Distribution versteht man ein Komplettpaket von Software, die nach den Bedürfnissen einer speziellen Zielgruppe vom Distributor zusammengefasst und teilweise angepasst wird. Am bekanntesten sind Linux Distributionen. Weitere Distributionen gibt es von BSD[4], DOS, etc. Heute gibt es hunderte von verschiedenen Distributionen. Unter Distrowatch (http://distrowatch.com) werden viele davon aufgelistet und kurz beschrieben.

1 Gib die Inhalte/Eckpunkte der GPL wieder
2 HA: Was genau versteht man unter dem Kernel
3 Kurzreferat über das GNU Projekt
4 HA: Begriff BSD klären

X Window System

Unter Gnu/Linux ist es nicht wie unter Windows oder Mac selbstverständlich mit einer graphischen Arbeitsumgebung zu arbeiten. Vielmehr muss Linux erst in die Lage versetzt werden, eine Desktopumgebung darstellen zu können. Dazu muss ein so genannter X-Server installiert werden, auf dem später ein Desktop installiert werden kann. Einige Distributionen haben dies standardmäßig schon in ihrem Paket geschnürt.

Desktopumgebung

Unter Linux ist man nicht wie unter Windows oder Mac an eine vorgegebene Desktopumgebung (Window Manager) gebunden. Der Nutzer hat die Möglichkeit mehrere Umgebungen auf dem X-Server aufzusetzen.

Bekannte Vertreter sind:

KDE (http://de.wikipedia.org/wiki/Kde)
GNOME (http://de.wikipedia.org/wiki/GNOME)
XFCE[5] (http://de.wikipedia.org/wiki/Xfce)
FLUXBOX (http://de.wikipedia.org/wiki/Fluxbox)

1.2 Ubuntu

Im deutschsprachigen Raum ist SUSE als Distribution mit KDE als Desktop Umgebung am häufigsten vertreten. Dennoch haben wir uns in der Schule für Ubuntu[6] mit GNOME als Desktop entschieden.

- Ubuntu basiert, wie viele andere Distributionen auch, auf der sehr stabilen Distribution *Debian*.
- GNOME ist zur Zeit unter Ubuntu etwas besser als KDE implementiert.
- Ubuntu hat im letzten Jahr den größten Zuwachs und steht bei den Distributionen an Nummer 1[7]
- Ubuntu bietet eine sehr sinnvolle Vorauswahl an Programmen für jeden Einsatzzweck an. Eine mühevolle Suche nach passender Software entfällt.

5 Ressourcen schonend
6 Kurzreferat über die Geschichte von Ubuntu
7 Quelle: http://distrowach.com

2 Ubuntu, Installation und Support im WWW

2.1 Gründe umzusteigen

Es sprechen viele Gründe für einen Umstieg von Windows auf Linux.

- Man hat unter Windows deutlich mehr mit Viren, Trojanern und sonstigen Sicherheitslücken zu tun[8].
- Windows-Software ist oft sehr teuer und es gibt kaum günstige Alternativen.
- Linux ist absolut kostenfrei, es stehen sofort nach der Installation über 18.000 kostenfreie Programme zur Verfügung.
- Der Gedanke hinter Linux ist auf ein miteinander und füreinander ausgerichtet. Bei kommerziellen Betriebssystemen steht der Gewinn an erster Stelle und nicht der User.
- Der Support im WWW für Linux ist riesig. Es gibt zu jedem Thema genügend Informationsmaterial.
- Bei kommerziellen Betriebssystemen liegt der Quellcode[9] nicht offen. Es kann nicht nachvollzogen werden, ob und welche Informationen über das www übertragen werden.
- Die eigene Qualifikation anderen gegenüber erhöhen.

2.2 Hardware: Linux ist nicht gleich Windows

Unter Linux ist es immer noch ein Problem neuste Hardware in ein System einzubinden. Das liegt nicht daran, dass andere Betriebssysteme das besser können, sondern an der weiten Verbreitung von diesen und an dem Lizenz Modell von Linux. Damit eine bestimmte Hardware z.B. eine Grafikkarte genutzt werden kann, muss das Betriebssystem in der Lage sein, mit der Hardware zu kommunizieren. Diese Kommunikation übernimmt spezielle Software, der Treiber. Für Linux wird Software meist unter der GPL veröffentlicht. Firmen haben verständlicherweise kein Interesse daran, ihre Betriebsgeheimnisse preis zugeben. Das bedeutet, dass Entwickler erst mühsam einen Treiber für Linux entwickeln müssen. Dies hat zur Folge, dass neuste Hardware von Linux nicht sofort, sondern mit leichter Verzögerung unterstützt wird. Einige Firmen bieten Linux-Treiber an. Oft sind dies Treiber allerdings an eine bestimmte Linux Version gebunden. Vor dem Kauf eines neuen Rechners, lohnt sich eine Recherche im Internet, ob die Hardware schon unterstützt wird. Ubuntu wird ca. alle 6 Monate mit einer neuen Version aufgelegt. Wer somit nicht zur neusten Hardware greift, hat gute Chancen, dass die eingesetzte Hardware unterstützt wird. Wenn es im ein oder anderen Fall auch noch manueller Installation bedarf.

8 Linux ist keinesfalls absolut sicher. Dennoch ist die Gefahr eines Virus, ausspionieren des Systems um ein vielfaches geringer als bei Microsoft Windows.
9 HA: Begriff klären

Live CD

Die Ubuntu CD ist gleichzeitig eine Live CD, d.h. der Computer bootet von CD und der Nutzer kann mit Linux arbeiten, ohne das auf dem PC irgendetwas installiert oder verändert wird. Das dauert zwar etwas länger, dafür kann getestet werde ob die Hardware des eingesetzten PCs von Ubuntu erkannt wird.

Anlaufstellen im www:

- http://www.kefk.net/Linux/Hardware/
- http://www.linuxhardware.de
- http://www.linux.org/hardware/
- http://www.tuxhardware.de/

2.3 Bezugsquellen Ubuntu

Am einfachsten bezieht man Ubuntu kostenlos über das Internet (http://www.ubuntu.com/). Wer über keine Flatrate verfügt, kann sich die CD per Post kostenlos schicken lassen. Keine Angst vor der Englischen Sprache. Die Installation und auch alles andere erfolgt in Deutsch. Über das www können noch die Server Version (keine Desktopumgebung und spezielle Serverprogramme), *kubuntu* (KDE als Desktop), *xubuntu* (XFCE als Desktop) und *edubuntu* (mit speziellen Programmen für Schüler) bezogen werden.

2.4 Die Ubuntu Community

Die Linux Gemeinde zeichnet eine hohe Hilfsbereitschaft der Nutzer untereinander aus. Wer Probleme hat kann fragen. Wer eines gelöst hat lässt andere daran teilhaben.

IRC Internet Relay Chat

Nicht jeder kennt das Chat System IRC. Hier treffen sich Menschen mit den unterschiedlichsten Interessen. Derzeit gibt es ca. 550.000 Chat-Räume (http://searchirc.com/network/freenode) in über 3600 IRC Netzwerken.

Für uns interessant ist das *freenode* Netzwerk (http://freenode.net/) in dem es über 4000 Räume gibt. In *freenode* treffen sich Nutzer, die sich hauptsächlich über Free und Open Source Software („*FOSS*“) unterhalten. Genau das was wir brauchen. Interessante Räume sind: *#ubuntu*, *#ubuntu-de*, *#ubuntu-de-treffpunkt*. Um am IRC teilnehmen zu können benötigt man einen Client Programm. Hier bieten sich Programme wie *Gaim* oder *Miranda* an. Bei beiden handelt es sich um multifunktions Clients, die eine

Vielzahl von Protokollen[10] beherrschen (ICQ, AIM, MSN, Yahoo, IRC, ...) und für alle gängigen Betriebssysteme erhältlich sind. IM IRC ist keine Registrierung notwendig. Man nimmt einfach daran teil. Probiert es aus, das Zuschauen ist sehr interessant und informativ.

Ubuntu Foren

Neben den verschiedensten Linux Foren gibt es natürlich Foren, die sich ausschließlich mit Fragen rund um Ubuntu beschäftigen. Da Ubuntu auf Debian basiert, können oft Informationen über Debian auf Ubuntu übertragen werden.

deutsch:

- http://forum.ubuntuusers.de/
- http://www.kubuntu.de/forum/forum.php
- http://www.ubuntuforum.de/
- http://ubunux.de/

englisch:

- http://www.ubuntuforums.org/
- http://kubuntuforums.net/

Anleitungen und Anlaufstellen

deutsch:

- http://www.ubuntuusers.de/ (offizielles deutsches Ubuntu Portal)
- http://wiki.ubuntuusers.de/Startseite (hier gibt's Anleitungen zu so ziemlich jedem Thema)
- http://ubuntuguide.org/wiki/Ubuntu_dapper_de (inoffizielles Handbuch / Installationsleitungen)
- http://www.kubuntu.de/ (Kubuntu Portal)
- http://www.galileocomputing.de/openbook/ubuntu/ (online Version eines Buches zu Ubuntu)
- http://de.wikibooks.org/wiki/Linux-Kompendium:_Ubuntu (Kompendium zu Ubuntu)
- http://verein.ubuntu-de.org/ (Ubuntu Verein)
- http://de.wikipedia.org/wiki/Ubuntu (Nützliche Hinweise)
- http://www.elyps.de/ (Homepage von Buchautor Markus Fischer, mit online Buch)

- http://packages.ubuntu.com/ (hier könnt ihr suchen, welche Software es wo gibt)

englisch:

- http://www.ubuntu.com/community (viele nützliche Links)

10 fürs Erste muss „Sprachen" reichen

2.5 Installation von Ubuntu

Unter http://wiki.ubuntuusers.de/Installation finden sich viele Hinweise und Anleitungen, die sich mit der Installation von Ubuntu, Kubuntu, Edubuntu, Server, ... befassen. Es gibt Tipps zur Partitionierung und Lösungsvorschläge zu evtl. auftretenden Problemen.

Installationsanleitung

Auf http://wiki.ubuntuusers.de/Ubuntu_Installation geht es nur um Ubuntu. Es wird detailliert und bebildert beschrieben wie man Ubuntu auf einem Rechner installiert.

Vorsicht

Ubuntu verträgt sich problemlos mit Windows. Wer Ubuntu parallel zu Windows betreiben will sollte dennoch einige Punkte beachten:

- Vor der Installation unbedingt persönliche Daten und Einstellungen sichern.
- Wer mit dem Begriff Partitionierung nichts anfangen kann sollte folgenden Artikel lesen http://wiki.ubuntuusers.de/Partitionierung. Es ist zu empfehlen eine Partition für */tmp*[11] anzulegen. Das */tmp* Verzeichnis dient als Müllablage für Nutzer und Programme. */tmp* wird bei jedem Systemstart geleert. Auf einer eigenen Partition kann das System nicht zugemüllt werden.
- Wer seine Windows Partition verkleinern will um für Ubuntu Platz (mindestens 5 GB) zu schaffen, sollte vorher unbedingt defragmentieren. Ein vorübergehendes Abschalten der Auslagerungsdatei und der Systemwiederherstellung ist zu empfehlen, aber nicht notwendig.
- Die größte Gefahr Daten zu löschen besteht bei fehlerhaftem Partitionieren der Festplatte. Achtung: Windows-Partitionen nicht formatieren und nicht löschen!
- Um Daten zwischen Windows und Linux tauschen zu können wird eine mit FAT32 (vfat) formatierte Partition benötigt. Es kann sein, das diese in der Datei */etc/fstab*[12] falsch eingebunden wird und somit schreibgeschützt ist. Folgende Änderung beseitigt das Problem:

  `/dev/hda7  /media/tausch  vfat  defaults  0  1`
  ändern in
  *`/dev/hda7  /media/tausch  vfat  defaults`**`,utf8,umask=007,gid=46`** *`0  1`*

Grub

In der Installationsanleitung[13] wird geschrieben, dass Grub im Master-Boot-Record (MBR) installiert wird.

HA: Was bedeutet das?

11 200 MB reichen. Möglich ist auch /tmp nach /var/tmp zu verlinken (wenn var eine eigene Partition hat)
12 Es wird festgelegt, welche Partitionen wie und wo im Verzeichnisbaum beim Systemstart eingebunden werden.
13 http://wiki.ubuntuusers.de/Ubuntu_Installation

2.6 Bezeichnung von Festplatten ***

Unter Windows werden Laufwerke mit Buchstaben bezeichnet (A:, C:, D:, ..., Z:). Dabei spielt es für Windows keine Rolle, ob es sich um eine Festplatte, oder um eine Partitionen auf einer Festplatte handelt. Unter Linux ist das anders. Linux unterscheidet sehr wohl, ob eine Festplatte, oder eine Partition vorliegt und an welchem Controller sie an der Hauptplatine angeschlossen ist.

Unter Linux befinden sich alle Laufwerke im Verzeichnisbaum (/). Die Laufwerke befinden sich im Ordner */dev*. Festplatten (auch CD-Rom Laufwerke) an IDE Controllern werden mit *hd*, SATA, SCSI und USB Sticks mit *sd* und Software-Raids[14] werden mit *md* (das erste mit *md0*) bezeichnet. Bei den Geräten spielt es eine Rolle an welchem Controller bzw. in welcher Reihenfolge sie angeschlossen sind.

hda: Master an IDE1	(/dev/hda)
hdb: Slave an IDE1	(/dev/hdb)
hdc: Master an IDE2	(/dev/hdc)
hdd: Slave an IDE2	(/dev/hdd)
sda: erste SCSI bzw. SATA Platte oder USB Stick	(/dev/sda)
sdb: zweite SCSI bzw. SATA Platte oder USB Stick	(/dev/sdb)
sdc: dritte SCSI bzw. SATA Platte oder USB Stick	(/dev/sdc)

Primäre Partitionen

Mit diesen Bezeichnungen kann Linux noch nicht arbeiten, da eine Aussage über die Partitionierung fehlt. Ursprünglich konnten Festplatten in nur maximal *vier* Partitionen (primäre Partitionen) unterteilt werden. Über die Zahlen 1-4 können diese (primären) Partitionen angesprochen werden.

D.h. die erste primäre Partition auf der Festplatte am ersten IDE-Controller über *hda1* (/dev/hda1), die zweite mit *hda2* (/dev/hda2), die dritte mit *hda3* (/dev/hda3) und die vierte mit *hda4* (/dev/hda4).

Erweiterte und logische Partitionen

Die Unterteilung in 4 Partitionen erwies sich schnell als zu gering. Das Problem wurde durch die Einführung von der erweiterten Partition gelöst. Anstelle einer primären Partition konnte nun eine erweiterte Partition anlegt werden, die wiederum in beliebig viele logische Partitionen unterteilt werden kann. Die mögliche Anzahl der logischen Partitionen hängt vom Betriebssystem ab. Die erste logische Partition erhält die Nummer 5, die zweite die 6, usw. Nach der Erstellung einer erweiterten Partition kann keine primäre mehr erstellt werden. Dies ist die Erklärung, weshalb die Nummerierung einer Festplatte oft Löcher aufweist. Meistens unterteilt man Festplatten in eine primäre und eine erweiterte Partition, die in mehrere logische Partitionen unterteilt wird[15].

14 Kurzreferat: Was versteht man unter einem RAID und welche Arten gibt es.

15 Erstelle eine Partitionierungstabelle einer Master IDE am zweiten Controller mit 1 primären und vier log. Partitionen. Wie lautet der Zugriff auf den USB Stick (es befindet sich keine SATA oder SCSI Platte im PC)?

Vorteile der Partitionierung

- Mehrere Betriebssysteme können auf einem PC installiert werden
- Bestimmten Bereichen kann eine eigene Partition zugewiesen werden.
  - Die Partition kann separat gesichert werden.
  - Trennung von Daten und Betriebssystem.
  - Eine Partition kann einem bestimmten Zweck zugewiesen werden und ist von anderen Teilen des Systems „sicher“.

2.7 Ubuntu ist installiert

Auf den ersten Blick könnte man meinen man hat ein abgeändertes Microsoft Windows vor sich. Dem ist absolut nicht so, obwohl sich in der Handhabung einige Dinge kaum bis gar nicht unterscheiden. Die in der Schule genutzte Software ist bis auf sehr wenige Ausnahmen auch für MS Windows erhältlich und unterscheidet sich in der Handhabung nicht. Deshalb besteht kein Grund hier auf diese Software einzugehen. Wichtiger ist es sich so schnell wie möglich mit dem Betriebssystem zurecht zu finden. Bitte bedenkt, die meisten von euch arbeiten schon seit Jahren mit Microsoft Windows. Erwartet nicht, dass ihr Linux innerhalb von zwei Wochen beherrscht. Gebt Linux eine Chance und arbeitet eine Zeit lang *nur* mit Ubuntu.

Microsoft Windows nicht löschen

Es gibt Gründe, weshalb man Windows nicht ganz vom Computer verbannen sollte:

- Wenn ihr spezielle Hardware nutzt, die von Linux noch nicht unterstützt wird.
- Ihr spielt sehr gerne und einige Spiele gibt es nur für Windows.
- Ihr benötigt Software, für die es unter Linux kein Ersatz gibt.

3 Es wäre schön root zu sein

3.1 Der normale Nutzer unter Linux

Der normale user unter Linux darf (fast) alle installierten Programme starten und sich an vielen Stellen im System umschauen. Ansonsten darf er nichts und das ist gut so. Es gibt keinen Grund weshalb ein user für seine tägliche Arbeit mehr Rechte erhalten sollte als unbedingt notwendig. Bei Microsoft ist das, zumindest bis Windows XP, anders. Jeder Nutzer ist dort standardmäßig Administrator, mit der Möglichkeit sich frei im System zu bewegen. Unter Linux heißt der Administrator *root*.

Vorteile ein Administrator bzw. root zu sein ***

Mir fällt nur einer ein, es ist einfach „sau bequem“. Leider aber auch super riskant.

Nachteile ein Administrator bzw. root zu sein ***

Es sprechen viele Gründe gegen die Nutzung des Systems als root. Zu Zeiten des Internets ist die Gefahr sehr groß, dass ein Eindringling, ein Virus, oder ein Trojaner genau die Rechte erhält, die der Nutzer hat dem solch eine Software unter geschoben wird. Handelt es sich bei dem Nutzer um einen user mit root Rechten, so stehen dem Schädling Tür und Tor offen. Bei einem normalen user beschränkt sich der Schaden auf den user selbst, das System bleibt voll funktionstüchtig. Zudem kann man als root einem System versehentlich, oder aus Unwissenheit heraus großen Schaden zufügen.

3.2 Es gibt kein root

Diese Aussage trifft so nicht ganz zu. Der allmächtige user root ist zwar vorhanden, ihm ist aber kein Passwort zugeordnet. Das führt dazu, dass man sich als root nicht einloggen kann[16]. Der bei der Installation angelegte user ist als einziger in der Lage root Rechte zu erlangen, oder anderen usern diese Fähigkeit zu gewähren. Es soll verhindert werden, dass man sich aus Bequemlichkeit als root einloggt und somit alle Sicherheitsaspekte außer Acht lässt. Für Nutzer anderer Distributionen ist dieser Umstand sehr gewöhnungsbedürftig.

16 Im Anhang befindet sich eine Anleitung um unter Ubuntu root nutzen zu können.

3.3 Root Rechte erlangen

root Rechte ***

Nur der bei der Installation angelegte user kann root Rechte erlangen bzw. diese anderen Nutzern gewähren. Nutzern root Rechte zu erteilen. Menüpunkt *System->Administration->Benutzer und Gruppen*. Dort kann unter *Eigenschaften->Benutzerrechte* das „*Durchführen von administrativen Aufgaben*“ selektiert werden.

Nur solche Benutzer dürfen Programme zum Verwalten des Systems unter Eingabe *ihres* Passwortes starten. Soll solch ein Programm aus der Konsole heraus gestartet werden, bzw. soll ein Befehl mit root Rechten ausgeführt werden, muss der Befehl *sudo* vorangestellt werden.

Z.B.:

● *sudo synaptic*	Programm zur Softwareverwaltung
● *sudo gedit /etc/apt/sources.lst*	Die Datei sources.lst darf nur mit root Rechten editiert werden
● *sudo cp -r /etc/* /home/sabath/etc*	Nur mit root Rechten können alle Dateien in /etc an eine andere Stelle kopiert werden.

Mein Account kann keine root Rechte erlangen

In einem solchen Fall ist es nicht notwendig sich immer erst auszuloggen und als Benutzer mit root Rechten einzuloggen.

Möglichkeit1:

In einem Terminal-Fenster durch die Eingabe von *su username*. Innerhalb des Fensters ist man nun als der angegebene Nutzer angemeldet.

Möglichkeit 2:

Sollte keine graphische Umgebung benötigt werden, kann über die Tastenkombination *Strg+Alt+[F1-F6]*[17] in eines der sechs zur Verfügung stehenden Pseudo-Terminals umgeschaltet werden, wo man sich erneut einloggen kann. *Alt+[F1-F6]* schaltet zwischen den Terminals hin und her[18]. *Alt+F7* bringt die grafische Umgebung zurück.

17 Eine Funktionstaste von F1-F6

18 Mehrere Nutzer sollen sich an eurem Rechner auf einem anderen Terminal einloggen. Prüft mit dem Befehl *w*, wer gerade eingeloggt ist.

4 Software installieren

4.1 Vorbereitung: /etc/apt/sources.list

Für Ubuntu gibt es weit über 18.000 frei erhältliche Programme, die unter der GPL veröffentlicht sind. Hinzu kommen noch Programme, die zwar frei erhältlich sind, deren Quelltext aber nicht veröffentlicht ist bzw. nicht verändert werden darf. Unter GNU/Linux darf nur Software verteilt werden, die der GPL unterliegt. Es gibt spezielle Server mit freier Software, die nicht der GPL unterliegt. Klingt kompliziert, ist es aber am Ende dieses Kapitels nicht mehr.

Das Zauberwort heißt *sources.lis*t (/etc/apt/sources.list). In dieser Datei wird eingetragen von welchen Servern Software bezogen werden darf. Als Standardeinstellung sind nur die vom Ubuntu-Team gepflegten Server aktiviert, d.h. sie sind *nicht* mit einem # Zeichen auskommentiert. Bei diesen Servern kann man davon ausgehen, dass die Software unter der GPL veröffentlicht, ausgiebig getestet, stabil und „sicher“ ist. Allerdings müsste man auf einige Programme (z.B. Flash Player, Microsoft Fonts, Java, ...) verzichten. Deshalb unterhalten Entwickler Server, die Software bereithalten welche nicht die strengen Kriterien der GPL erfüllen. Die so genannten Universe- und Multiverse-Repository bzw. Backports[19]. Ubuntu übernimmt bei dieser Software keine Garantie im Bezug auf Sicherheit und Stabilität. Zugriff auf diese Server erhält man, indem die Kommentarzeichen (#) zu Beginn der entsprechenden Zeile entfernt werden. Alle bisher genannten Server unterhalten Bereiche, in denen Sicherheits-Updates bereitgestellt werden. Deshalb nicht vergessen auch diese zu aktivieren.

Eine Aktivierung von Servern darf nur ein user mit root Rechten vornehmen. Entweder über das Programm *synaptic*, oder durch Aufruf von *sudo gedit /etc/apt/sources.list*

Seit Ubuntu 6.06 unterhält die Firma Canonical (welche hinter Ubuntu steht) ein Repository mit frei zur Verfügung stehender kommerzieller Software wie z.B. der Realplayer. Mit der Zeile

```
deb http://archive.canonical.com/ubuntu dapper-commercial main
```

in */etc/apt/sources.lst* kann dieses Repository hinzugefügt werden. Nach jeder Änderung muss die Datei neu eingelesen werden. Das funktioniert unter *synaptic*[20], oder durch den Aufruf *sudo aptitude update*.

19 Bitte beide aktivieren und die entsprechenden Update Server
20 Siehe 4.3

4.2 Abhängigkeiten ***

Bevor es an das Installieren von Software geht, noch ein paar nützliche Tipps. Nicht jedes Programm bringt alle nötigen Bestandteile mit. Bei der Entwicklung von Software greifen die Entwickler auf schon vorhandene Bestandteile zurück um nicht vorhandenes neu programmieren zu müssen. Solche Programmteile stehen in so genannten Bibliotheken bzw. Libraries zur Verfügung. Ein Programm kann nur funktionieren wenn alle benötigten Libraries zur Verfügung stehen. Das Programm ist von diesen Libraries *abhängig*. Sind diese Abhängigkeiten nicht erfüllt funktioniert das Programm nicht. Genau hier liegt die große Stärke von auf Debian basierenden Distributionen. Debian verfügt über ein ausgeklügeltes System um bei tausenden installierten Programmen (es wird auch gerne von Paketen gesprochen) alle Abhängigkeiten zu erfüllen.

Mann kann sich ausmalen was passiert, wenn ein Programm A z.B. die Library *libacl.la* in einer bestimmten Version benötigt und das Programm B die selbe Library in einer älteren Version. Es kommt zum Konflikt und das System kann diesen Konflikt nicht auflösen. Das kann unter Umständen zu einem instabilen System führen. Bei auf Debian basierende Distributionen ist solch ein Fall nahezu ausgeschlossen.

4.3 Unter Gnome

Die Aufgabe die Datei */etc/apt/sources.lst* zu verwalten und Software zu installieren bzw. zu deinstallieren übernimmt unter Gnome das Programm *synaptic*. Die Bedienung ist intuitiv. Achtung, nach jeder Änderung der Datei */etc/apt/sources.lst* ist diese neu einzulesen.

Ein Programm/Paket kann *entfernt* bzw. *vollständig entfernt* werden. Beim Entfernen bleiben Konfigurationsdateien erhalten, während beim vollständigen Entfernen alles gelöscht wird.

4.4 Im Terminal / in der Konsole

aptitude

Auf der Konsole verrichtet das Programm *aptitude*[21] (Aufruf: *sudo aptitude*) die Arbeit von *synaptic* unter Gnome. Die Datei */etc/apt/sources.lst* kann mit aptitude nicht bearbeitet werden. Hierzu eignet sich *mc*[22], oder vi[23]. Aptitude ist ein mächtiges Werkzeug. Die Bedienung ist gewöhnungsbedürftig.

Beispiele:

21 Benutzerhandbuch: http://people.debian.org/~dburrows/aptitude-doc/en/
22 Siehe Kapitel 7.1
23 Siehe Kapitel 7.1

aptitude search Programm bzw. Suchstring	sucht ein Programm
aptitude update	liest die *sources.lst* neu ein
aptitude upgrade	aktualisiert Programme wenn möglich
aptitude remove Programm	entfernt ein Programm
aptitude purge Programm	entfernt auch die Konfigurationsdateien
aptitude -V Programm	anzeigen welche Version installiert ist

Sind bei aptitude Abhängigkeiten nicht erfüllt, werden die benötigten Libraries automatisch mit installiert[24].

Weitere Möglichkeiten Software zu installieren

Es gibt noch weitere mächtige Werkzeuge um Software zu installieren, deinstallieren und zu konfigurieren. Es handelt sich um Programme *apt-get* und *dpkg,* auf die hier nicht weiter eingegangen werden soll[25].

4.5 Starten von Programmen ***

Solange sich die Programme im Menü befinden funktioniert alles wie unter Windows. Gefällt die Struktur des Menüs nicht, kann es mit dem Programm *alacarte* konfiguriert werden. Es befindet sich im Menüpunkt *Zubehör*, oder ein Start über die Befehlszeile mit *Alt+F2* und Eingabe von alacarte.

Befehlszeile öffnen

Mit *Alt+F2* öffnet sich unter Linux eine Befehlszeile. Dort gilt es dann den Namen des gewünschten Programms einzugeben[26]. Linux hilft bei dem Programmnamen wo es nur geht. Ist für Linux der Name eindeutig, wird dieser automatisch ergänzt.

Achtung: Im Gegensatz zu Windows unterscheidet Linux bei der Groß- und Kleinschreibung. Ein Start von *nautilus* funktioniert, wohingegen die Eingabe von *Nautilus* ins Leere läuft.

Im Terminal

Identisch zu der Befehlszeile verhält sich der Start aus einem Terminal-Fenster. Abhängig vom Programm kann es sein, dass dieses Terminal-Fenster nach dem Starten eines Programms für die Zeitdauer in der das Programm geöffnet ist nicht nutzbar ist.

Beispiel: Start von Gimp[27] über ein Terminal-Fenster.

24 Installiere mc von der Kommandozeile, suche nach Programmen mit sun im Paketnamen und installiere sun-java5-bin1, sun-java5-fonts, sun-java5-jdk, sun-java5-jre, sun-java5-plugin.
25 https://help.ubuntu.com/community/SoftwareManagement
26 Starte: nautilus, gedit, synaptic, gnome-terminal und gaim
27 *gimp* eingeben

Über Doppelklick

Unter Linux gibt es keine .exe bzw. .msi Dateien. Wird ein Programm mittels synaptic oder aptitude installiert, befindet es sich entweder im Menü. Alternativ kann es über den Namen im Terminal gestartet werden. Bei Programmen, die manuell installiert wurden, liegt meistens eine Anleitung bei, oder man sucht nach einer ausführbaren Datei (oft mit der Endung .sh)[28].

Derzeit kann Nvu in deutsch (http://nvu-composer.de/) nicht über Ubuntu installiert werden[29]. Manuell sind folgende Schritte durchzuführen:

1. Die GTK Version für Linux downloaden nach z.B. */home/klaus/NAME-DER-DATEI.tar.gz*
2. *cd /opt/* (in das Verzeichnis /opt wechseln)
3. *sudo tar -xfvz /home/klaus/NAME-DER-DATEI.tar.gz*
   (die Datei wird an die aktuelle Position entpackt. x = entpacken, f = file / aus Datei, v = verbose / mit Rückmeldungen) und z = .gz gepackt.
4. *sudo mv /opt/nvu-1.0/ /opt/nvu* (umbenennen des Ordners nvu-1.0 in nvu)
   - *sudo /opt/nvu/nvu* (Einmaliger Aufruf mit root Rechten über den kompletten Pfad)
5. */opt/nvu/nvu* (so wird das Programm gestartet)
6. Mit *alacarte* einen Menüeintrag erstellen. Als Befehl */opt/nvu/nvu* eingeben (Wenn ihr Nvu im selben Verzeichnis wie hier in diesem Skript installiert habt).

4.6 Ubuntu und Multimedia

Viele Multimediaanwendungen entsprechen nicht der GPL. Sie sind an andere Lizenzmodelle gebunden bzw. werden von Firmen[30] vertrieben. Z.B. Flash, Shockwave, Java, Realplayer und DVD Player. Diese müssen unter Linux erst nachträglich eingebunden werden.

Der einfache Weg

EasyUbuntu (http://easyubuntu.freecontrib.org/) und Automatix (http://www.getautomatix.com/) bieten beide die Möglichkeit schnell und unkompliziert diese fehlende Funktionalität nachträglich einzubinden.

Der komplizierte Weg

Wer genau wissen möchte welche Pakete installiert werden, oder mehr über Installationsroutinen unter Linux erfahren will, der muss sich im Internet nach entsprechenden Informationen umtun. Ein guter Start ist: http://www.ubuntu.com/community.

mp3, Java, Win codecs, DVD, iTunes, Flash,... (https://help.ubuntu.com/community/RestrictedFormats)

Realplayer, Java und Flash (http://linuxwiki.de/Ubuntu)

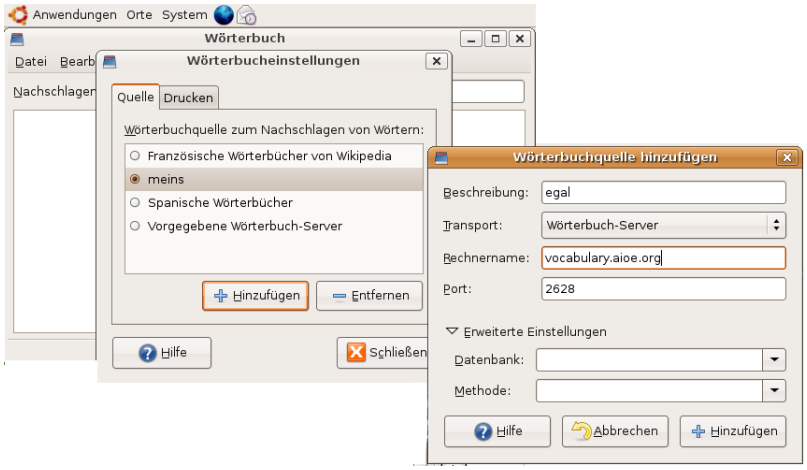
28 Was ganz entfernt mit einer .exe Datei unter Windows verglichen werden kann.
29 Wer die englische Version installiert hat bitte Hinweise auf nvu-composer.de beachten
30 meist als kostenlose Plugins

4.7 Microsoft Schriftarten

Um vorhandene MS Office Dokumente unter OpenOffice nutzen zu können, werden die Schriftarten von Microsoft benötigt[31]. Das erledigt das Paket *msttcorefonts*.

4.8 Panels erweitern

Panels können mit nützlichen Funktionen oder Hilfsprogrammen erweitert werden. Mit der rechten Maustaste auf den Panel klicken und „*Zum Panel hinzufügen*“ anwählen. Danach kann aus kleineren Hilfsprogrammen gewählt werden. Z.B. das Programm *Wörterbuchsuche*, welches leider nur auf englisch Wörter reagiert. Durch hinzufügen eines neuen Servers (*vocabulary.aioe.org*) lässt sich dies ändern.



Wer es bis hier her geschafft hat, sollte auf dem Computer problemlos arbeiten können. In der Schule ist alles etwas anders aufgebaut und die Voraussetzungen sind andere. Hier nutzen ca. 1000 Schüler die Computerräume. Prinzipiell ist das kein Problem. Was aber, wenn Dateien getauscht bzw. vor neugierigen Blicken geschützt werden sollen.

31 Es müssen die Universe und Mulitverse-Repository aktiviert sein.

5 Zugriffsrechte unter Linux ***

Niemand hat es gerne, wenn andere in seinen Sachen herum wühlen. Unter Linux ist es Standard, dass kein user die Dateien eines anderen ändern darf.

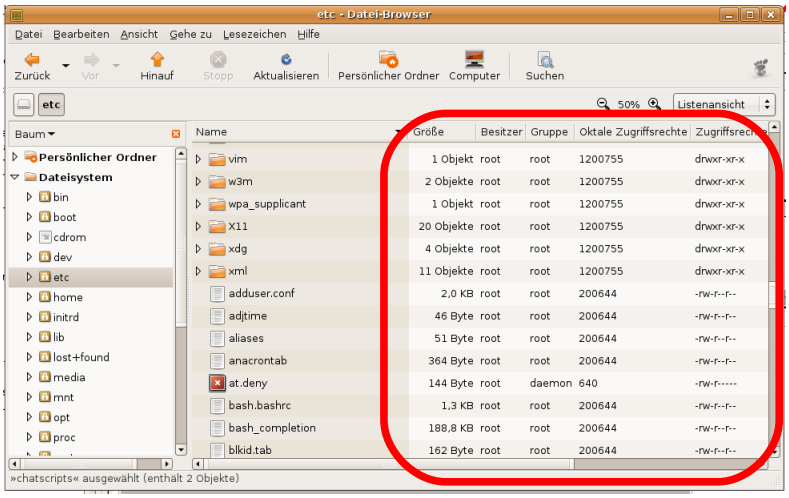
5.1 Grundlagen

Jeder Benutzer erhält unter Linux einen eindeutigen Benutzernamen, diese wird intern bei Ubuntu[32] mit einer Zahl ab 1000 repräsentiert. Der erste user erhält bei Ubuntu die user-id (*uid*) 1000, der zweite 1001, etc. Gleichzeitig gibt es unter Linux Gruppen. In einer Gruppe können user zusammengefasst werden. Unter Ubuntu erhält jeder user seine eigene Gruppe. Gruppen werden intern wieder als Zahlen dargestellt. Der erste angelegte user erhält die Gruppen-id (*gid*) 1000, die zweite gid lautet 1002, etc. Als drittes gibt es noch alle anderen (*others*), den Rest der Welt. Anhand dieser drei Merkmale: user(u), group(g), others(o) kann nun unter Linux differenziert angegeben werden, welcher user bzw. welche Gruppe was auf dem System darf. Z.B. eine Datei lesen, verändern oder gar ausführen bzw. ein Verzeichnis betreten?

In der Schule sind alle Schüler der gleichen Gruppe „*Schueler*“ zugeordnet.

5.2 Anzeigen der Zugriffsrechte

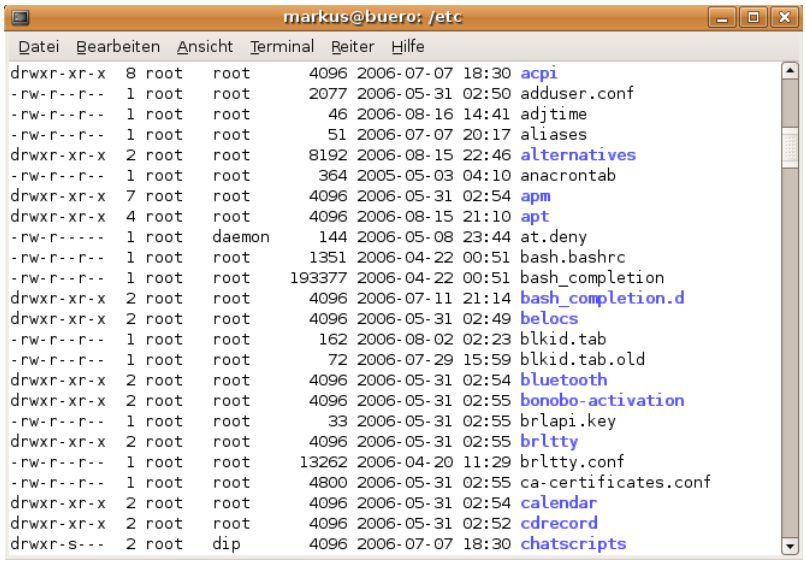
Im Dateibrowser Nautilus können die gewünschten Informationen über *Besitzer*, *Gruppe, oktale Zugriffsrechte* und *Zugriffsrechte* anzeigen werden. Menüpunkt Ansicht->anzuzeigende Spalten.


etc - Datei-Browser
Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe
Zurück Vor Hinauf Stopp Aktualisieren Persönlicher Ordner Computer Suchen
etc
50% Listenansicht
Baum
Persönlicher Ordner
Dateisystem
bin
boot
cdrom
dev
etc
home
initrd
lib
lost+found
media
mnt
opt
proc
Name Größe Besitzer Gruppe Oktale Zugriffsrechte Zugriffsrec
vim 1 Objekt root root 1200755 drwxr-xr-x
w3m 2 Objekte root root 1200755 drwxr-xr-x
wpa_supplicant 1 Objekt root root 1200755 drwxr-xr-x
X11 20 Objekte root root 1200755 drwxr-xr-x
xdg 4 Objekte root root 1200755 drwxr-xr-x
xml 11 Objekte root root 1200755 drwxr-xr-x
adduser.conf 2,0 KB root root 200644 -rw-r--r--
adjtime 46 Byte root root 200644 -rw-r--r--
aliases 51 Byte root root 200644 -rw-r--r--
anacrontab 364 Byte root root 200644 -rw-r--r--
at.deny 144 Byte root daemon 640 -rw-r-----
bash.bashrc 1,3 KB root root 200644 -rw-r--r--
bash_completion 188,8 KB root root 200644 -rw-r--r--
blkid.tab 162 Byte root root 200644 -rw-r--r--
»chatscripts« ausgewählt (enthält 2 Objekte)

32 nicht bei allen Distributionen beginnen die user bei 1000

Im Terminal

Min *ls -l* kann man sich den Inhalt des aktuellen Verzeichnisses mit Zugriffsrechten anzeigen lassen.


markus@buero: /etc
Datei Bearbeiten Ansicht Terminal Reiter Hilfe
drwxr-xr-x 8 root root 4096 2006-07-07 18:30 acpi
-rw-r--r-- 1 root root 2077 2006-05-31 02:50 adduser.conf
-rw-r--r-- 1 root root 46 2006-08-16 14:41 adjtime
-rw-r--r-- 1 root root 51 2006-07-07 20:17 aliases
drwxr-xr-x 2 root root 8192 2006-08-15 22:46 alternatives
-rw-r--r-- 1 root root 364 2005-05-03 04:10 anacrontab
drwxr-xr-x 7 root root 4096 2006-05-31 02:54 apm
drwxr-xr-x 4 root root 4096 2006-08-15 21:10 apt
-rw-r----- 1 root daemon 144 2006-05-08 23:44 at.deny
-rw-r--r-- 1 root root 1351 2006-04-22 00:51 bash.bashrc
-rw-r--r-- 1 root root 193377 2006-04-22 00:51 bash_completion
drwxr-xr-x 2 root root 4096 2006-07-11 21:14 bash_completion.d
drwxr-xr-x 2 root root 4096 2006-05-31 02:49 belocs
-rw-r--r-- 1 root root 162 2006-08-02 02:23 blkid.tab
-rw-r--r-- 1 root root 72 2006-07-29 15:59 blkid.tab.old
drwxr-xr-x 2 root root 4096 2006-05-31 02:54 bluetooth
drwxr-xr-x 2 root root 4096 2006-05-31 02:55 bonobo-activation
-rw-r--r-- 1 root root 33 2006-05-31 02:55 brlapi.key
drwxr-xr-x 2 root root 4096 2006-05-31 02:55 brltty
-rw-r--r-- 1 root root 13262 2006-04-20 11:29 brltty.conf
-rw-r--r-- 1 root root 4800 2006-05-31 02:55 ca-certificates.conf
drwxr-xr-x 2 root root 4096 2006-05-31 02:54 calendar
drwxr-xr-x 2 root root 4096 2006-05-31 02:52 cdrecord
drwxr-s--- 2 root dip 4096 2006-07-07 18:30 chatscripts

Das entspricht in etwa der Anzeige in Nautilus aus, doch was soll es bedeuten? Wie bereits erwähnt, gibt es drei Arten von Zugriffsrechte:

- lesen (read, r)
- schreiben (write, w)
- ausführen, Verzeichnis betreten (execute, x)

Diese können vergeben werden an: den Besitzer (user, u), die Gruppe (group, g) oder alle anderen (others, o)

Zugriffsrechte						
	lesen		**schreiben**		**ausführen**	
oktal	**4**		**2**		**1**	
Eigentümer (user, u)	r	-	w	-	x	-
Gruppe (group, g)	r	-	w	-	x	-
Andere (others,o)	r	-	w	-	x	-

z.B. etc drwx r-x r-x root root

user group others

Bei etc handelt es sich um ein Verzeichnis, das dem user root gehört, Mitglied in der Gruppe root. Mitglieder der Gruppe root dürfen das Verzeichnis „betreten“, aber nicht ändern. Das gilt auch für alle anderen Nutzer.

Das *d* zeigt an, dass es sich um ein Verzeichnis handelt, bei einer Datei ist an dieser Stelle ein **–**
Danach kommen drei Blöcke je drei Zeichen, die die Zugriffsrechte auf dieses Verzeichnis bzw. die Datei angeben.

5.3 Zugriffsrechte mit Nautilus ändern

Es genügt mit der rechten Maustaste auf das Verzeichnis oder die Datei zu klicken. Besitzt man Schreibrechte können die Rechte nach eigenen Wünschen einstellt werden. Der Nachteil ist, dass dieser Vorgang für jedes Verzeichnis bzw. jede Datei wiederholt werden muss. Bei mehreren Dateien kann dies dauern. Auf die „B*esonderen Attribute*“ wird an dieser Stelle nicht eingegangen.

Vergebt so wenig Rechte wie möglich!!!!

5.4 Zugriffsrechte mit chmod ändern

Im Terminal können die Zugriffsrechte mit dem Befehl *chmod* geändert werden.

Rechte ändern mittels symbolischen Ausdruck

chmod [ugoa] [+ - =] [rwx] Verzeichnis/Dateiname

[ugoa] : u = user g = group o = others oder a = all

[+ - =] : + = hinzufügen - = entfernen = → zuordnen

Beispiele:	***alle Dateien/Verzeichnisse gehören dem Nutzer klaus***
/home/klaus/privat drwx r-x r-x	*chmod go-x /home/klaus/privat*
soll vor den Blicken anderer geschützt werden	da *privat* ein Verzeichnis ist[33] muss allen außer dem user selbst das Recht in das Verzeichnis zu wechseln (x) entzogen werden (-)
/home/klaus/tausch/alle - rwx r- - r- -	*chmod go+w /home/klaus/tausch/alle*
jeder soll ändern dürfen	es handelt sich um eine Datei zu der die Gruppe und alle anderen Schreibrechte benötigen
/home/klaus/tausch/alle	*chmod go=r /home/klaus/tausch/alle*
klaus hat vergessen wie er sich die Rechte anzeigen lassen kann. Er möchte aber die Datei für alle anderen auf schreibgeschützt setzen.	die Rechte werden gesetzt. Die bisherigen Einstellungen interessieren nicht. Mit a=r wäre die Datei auch für *klaus* schreibgeschützt

Es handelt sich hier nur um Lösungsvorschläge. Es gibt mehrere Möglichkeiten die Aufgaben zu erfüllen.

33 das d steht für directory

Differenzierte Einstellungen können mit Komma getrennt angegeben werden: *chmod u=rw,go=r*[34] *datei.txt*. Soll nicht nur eine Datei, sondern mehrere Dateien in einem Verzeichnis geändert werden[35], kann mit so genannten *Wildcards* gearbeitet werden. Wildcards sind Platzhalter/Joker für Zeichen.

* 0 bis beliebig viele Zeichen
? genau ein Zeichen

*chmod a=rw /home/klaus/**	ändert alle Dateien in einem Verzeichnis für alle auf lesen und schreiben
*chmod o= /home/klaus/**	für die anderen werden alle Rechte entzogen
chmod a=rw /home/klaus/.txt*	ändert alle Dateien mit der Endung .txt in einem Verzeichnis
chmod a=rw /home/klaus/.htm**	ändert alle Dateien mit der Endung .htm und html in einem Verzeichnis
chmod a=rw /home/klaus/.htm?*	ändert alle Dateien mit der Endung und html[36] in einem Verzeichnis

Soll der Befehl auch für alle Unterverzeichnisse gelten, so gibt es die schöne aber „gefährliche“ Option -R.
*chmod -R a=rw /home/klaus/** alle Dateien in allen Unterverzeichnissen erhalten die angegebenen Zugriffsrechte. Die Option -R ist sehr mächtig. Man stelle sich folgendes vor: *sudo chmod -R a=rwx /* alle Dateien auf dem PC erhalten die identischen Zugriffsrechte. Das war es dann, viel Spaß beim Neuinstallieren des Systems. Ein weiterer Grund, weshalb es nicht so sinnvoll ist mir root Rechten unterwegs zu sein.

Wenn ihr sehen wollt, was chmod gerade tut, ergänzt den Aufruf mit der Option -v.
Z.B. *chmod -Rv a=rw /home/klaus/alle*.*

Rechte ändern mittels Oktalzahl

Die Rechtemaske für Eigentümer, Gruppe und für die anderen bestehen jeweils aus drei Spalten (rwx), oder besser gesagt aus drei Bits. Ein gesetztes Bit (1) steht für ein erteiltes Recht und die 0 für ein verweigertes Recht. Eine Oktalzahl ist eine Zahl zur Basis 8^1. Alle Ziffern können mit drei Bits dargestellt werden. 000 = 0, 001 = 1, 010 = 2, 011 = 3, ..., 111 = 7. Im Binärsystem entspricht das hinterste Bit $2^0 = 1$, das mittlere Bit $2^1 = 2$ und das vorderste $2^2 = 4$. Dies entspricht den Oktalzahlen angegeben in der Tabelle Zugriffsrechte aus 5.2. Somit lassen sich die Rechte nicht nur mit einem symbolischen Ausdruck ändern, sondern auch mit Oktalzahlen.

```
rwx = 4+2+1 = 7        r-x = 4+0+1 = 5        r-- = 4+0+0 = 4
-wx = 0+2+1 = 3        -w- = 0+2+0 = 2        --x = 0+0+1 = 1
```

chmod 777 test.txt = chmod a=rwx test.txt
chmod 766 test.txt = chmod u=rwx,go=wx test.txt
chmod 600 test.txt = chmod u=rw,go-rwx test.txt
chmod 111 test.txt = chmod a=rwx test.txt
chmod 755 /test = chmod u=rwx,go=rx /text
chmod 644 test.txt = chmod u=rw,go=r test.txt

} Standardeinstellung für neue Verzeichnisse bzw. Dateien

34 kein Leerzeichen nach dem Komma
35 Reguläre Ausdrücke funktionieren auch: chmod +whatever $(ls | egrep 'Ausdruck')
36 nicht .htm, da das ? genau für ein Zeichen steht.

6 Linux konfigurieren

Wir werden in diesem Kapitel Linux nicht komplett konfiguriert, es soll vielmehr ein Überblick gegeben werden, wie Linux aufgebaut ist, an welcher Stelle systemweite und wo individuelle Änderungen vorgenommen werden.

6.1 Textdateien ***

Unter Linux werden alle Einstellungen über Textdateien vorgenommen. Das ist ein riesiger Vorteil gegenüber anderen Betriebssystem. Änderungen können kommentiert und somit wieder jederzeit rückgängig gemacht werden. Änderungen mit Hinweisen und Kommentaren zu versehen sei jedem ans Herz gelegt. Schon nach kurzer Zeit weiß man nicht mehr was man an welcher Stelle geändert hat. Jede Zeile, die mit einem # beginnt wird als Kommentar behandelt und vom System nicht ausgewertet. D.h. Änderungen können problemlos mit Hinweisen versehen werden.

Unter Debian besteht keine Gefahr, dass vom Nutzer gemachte Änderungen vom System einfach und ohne Nachfrage überschreiben werden.

6.2 Systemweite Konfiguration

Änderungen die sich auf das gesamte System auswirken, können nur von einem Nutzer mit root Rechten durchgeführt werden. Alle nötigen Dateien befinden sich im */etc* Verzeichnis[37]. D.h. Wenn etwas konfiguriert werden soll, dann gilt es dort die entsprechende Textdatei zu finden.

Gnome

Mit sudo *gedit /etc/pfad-zur-datei/einstellen.cfg* kann eine Datei mit root Rechten zum Editieren geöffnet werden.

Konsole

vi als Editor ist auf jedem Linuxsystem vorhanden. Wenn nichts funktioniert, vi läuft. Sollte der Midnight Commander installiert sein ist er vi vorzuziehen, da seine Bedienung einfacher ist.

37 Referat: Was ist mit all den anderen Verzeichnissen?

6.3 Benutzerspezifische Konfiguration

Passt ein Nutzer das System an seine Wünsche an, werden diese Einstellungen auch in Textdateien gespeichert. Entweder in versteckten Verzeichnissen, oder in versteckten Konfigurationsdateien. Versteckte Dateien/Verzeichnisse beginnen mit einem . (Punkt) und werden vom System standardmäßig nicht angezeigt. Unter Nautilus können die versteckten Verzeichnisse/Dateien mit *Strg.+H* und in der Konsole mit *ls -a* angezeigt werden.

6.4 Datensicherung

Unter Linux sollte unbedingt das Verzeichnis */etc* gesichert werden, es ist nicht sehr groß ca. 10 MB. Wurden größere Änderungen am System vorgenommen ist eine erneute Sicherung fällig.

Die Daten der Nutzer befinden sich alle unterhalb des */home* Verzeichnisses. Das */home* Verzeichnis sollte mit all seinen Unterverzeichnissen in regelmäßigen Abständen gesichert werden.

6.5 Fehlersuche

Zunehmend gibt es Programme, die eine Änderung der Konfigurationsdateien abnehmen und die entsprechende Textdatei editieren. Das ist deutlich bequemer, hat aber den Nachteil, dass nicht hervorgeht, was das Programm in welcher Datei geändert hat. Zu empfehlen ist eine direkte Änderung der Konfigurationsdateien. Anfangs muss mehr gesucht werden. Doch es bleibt der große Vorteil alle Änderungen in der Datei protokollieren zu können. Mit der Zeit lernt man so sein System richtig gut kennen.

Log Dateien

Wichtige Systemmeldungen werden in log Dateien geschrieben. Diese werden gespeichert in /var/log. Bei Problemen findet sich dort meist ein hilfreicher Hinweis. Hat ein Programm keine eigene log Datei findet sich hoffentlich ein Hinweis in */var/messages* bzw. */etc/dmesg* (Meldungen des Bootvorgangs).

Start über ein Terminalfenster

Sollte ein Programm nicht über das Menü zu starten sein. Hilft ein Start über ein Terminal-Fenster. Dort werden Meldungen über den Grund der Arbeitsverweigerung angegeben.

6.6 Nichts geht mehr

Das gibt es „nicht“. Irgendetwas geht immer. Doch was tun wenn das System sich nicht mehr starten lässt? Mit etwas Glück gelingt der login auf einer Pseudo-Terminal. Dort kann mittels vi bzw. mc die letzte fatale Änderung rückgängig gemacht werden. Wohl dem, der die Änderungen kommentiert hat. Eine ausgedruckte Kopie der Datei */etc/fsta*b kann sehr hilfreich sein. Dort ist angegeben wie die Festplatten und Partitionen eingebunden sind. Ist selbst ein login auf der Konsole nicht möglich hilft ein Start von CD mit Knoppix (http://www.knopper.net/knoppix/). Hierbei handelt es sich um eine Linuxdistribution als Live CD. Der PC startet auf jeden Fall und Änderungen an den Konfigurationen können vorgenommen werden.

Wer tiefer in Linux einsteigen will kommt nicht um eine Bedienung des Systems von der Konsole aus herum. Im folgenden Kapitel werden einige Befehle, Techniken und Programme vorgestellt werden, die helfen mit dem System sicherer umgehen zu können.

7 Editoren und einige Tipps

7.1 Editoren

Der Midnight Commander

Mit dem *mc* (Midnight Commander) steht auf der Konsole ein Programm zur Verfügung, dass den user bei vielen Aufgaben unterstützend zur Seite steht. Sich im Dateisystem bewegen, Dateien verschieben, kopieren und editieren stellen für den *mc* kein Problem dar. Allerdings muss es erst installiert werden.

Der Editor vi/vim ***

vi = visual und vim = vi improved

Dieser Editor steht auf jeder Linuxdistribution zur Verfügung. Das Arbeiten mit vi ist für einen Neuling gewöhnungsbedürftig. Die Grundfunktionen sollten jedoch bekannt sein, um nötige Änderungen an Konfigurationsdateien durchführen zu können, auch wenn einmal keine graphische Umgebung zur Verfügung steht. Hier ein kleiner Überblick über die Funktionen von vi.

vi /etc/apt/sources.lst öffnen einer Datei in vi. Ist die Datei nicht vorhanden, wird sie angelegt[38].

vi +/Zeichenkette Dateiname öffnet die Datei und der Curser geht zum ersten Vorkommen der *Zeichenkette*.

Vi unterscheidet drei verschiedenen Modi, in denen gearbeitet werden kann.

1. *Der Befehlsmodus*
   In diesem Modus startet vi. Die Datei kann noch nicht editiert werden! Jede Eingabe wird als Befehl zum Bearbeiten der Datei interpretiert.

2. *Der Eingabemodus*
   Mit z.B. der Taste *i* (insert) gelangt man in diesen Modus. Die Datei kann nun editiert werden. Mit *Esc* gelangt man in den Befehlsmodus zurück.

3. *Die Kommandozeile*
   Ausgehend vom Befehlsmodus durch Eingabe eines : (Doppelpunktes). In diesem Modus kann die Datei manipuliert werden (speichern, speichern unter, Schließen, nächste Datei öffnen).

38 neue leere Dateien können schneller mit dem Befehl *touch Dateiname* angelegt werden.

Der Befehlsmodus

yy	kopiert die aktuelle Zeile
ny	kopiert n+1^Zeilen
ya	kopiert ein Wort rechts vom Cursor
yb	kopiert ein Wort links vom Cursor
:reg	zeigt den kopierten Inhalt an
P	fügt kopierten Inhalt vor dem Cursor ein
p	fügt den kopierten Inhalt nach dem Cursor ein
dd	löscht die aktuelle Zeile
dw	löscht ein Wort
/muster	sucht das *muster* im Text
/	sucht weiter
?	sucht rückwärts
?muster	sucht das *muster* rückwärts
:s/alt/neu	sucht und ersetzt alt durch neu (nur den ersten Treffer in der aktuellen Zeile)
:s/alt/neu/g	sucht und ersetzt alt durch neu in der gesamten aktuellen Zeile
:%s/alt/neu	sucht und ersetzt im gesamten Dokument den ersten Treffer pro Zeile
:%s/alt/neu/g	sucht und ersetzt im gesamten Dokument

Der Eingabemodus (ausgehend vom Befehlsmodus)

i	*** Einfügen links von Cursor
I	Einfügen am Zeilenanfang
A	Einfügen am Zeilenende
o	neue Zeile hinter der aktuellen einfügen
O	neue Zeile vor der aktuellen einfügen
s*Text*	ersetze ein Zeichen durch „Text“
S*Text*	ersetze die ganze Zeile durch „Text“
cw*Text*	ersetze das Wort durch „Text“

Die Kommandozeile

:w	*** speichert die Datei
:q	*** beendet das Programm
:q!	*** beendet das Programm. Änderungen werden verworfen
:w!	*** speichert trotz Schreibschutz
:w name	speichert die Datei unter dem angegebenen Namen[39]

39 den Pfad mit angeben, ansonsten landet die Datei in . (im aktuellen Verzeichnis)

:wq *oder* :x	s*** peichert die Datei und beendet das Programm
:e name	öffnet die angegebene Datei
:e! name	öffnet die Datei erneut und verwirft alle Änderungen
:next	nächste Datei (falls vi beim starten mehrere Dateien angegeben wurden)
:prev	vorherige Datei

7.2 Die Tabulatortaste ***

Zum Glück nutzt Linux die Tabulatortaste nicht nur als nützliches Feature beim Eingeben von Text. Auf der Konsole wird man sich nach kürzester Zeit fragen, was man ohne sie machen würden. Sie hat zwei Funktionen:

Befehle ergänzen:

Bei den vielfältigen Möglichkeiten von Linux läuft man schnell Gefahr die korrekte Schreibweise von Befehlen zu vergessen. Zumal Linux Wert auf die Groß- und Kleinschreibung legt. Kein Problem: Die Anfangsbuchstaben eines Befehls tippen, danach die Tabulatortaste und siehe da, Linux ergänzt den Befehl. Sollte eine Zuordnung nicht eindeutig möglich sein, listet ein zweites Betätigen der Tabulatortaste alle Möglichkeiten auf.

Pfade ergänzen:

Es ist durchaus schwierig, den korrekten Pfad zu einer Datei anzugeben. Hier steht die Tabulatortaste ebenso hilfreich zur Seite wie beim Ergänzen von Befehlen. Einmaliges Drücken ergänzt den Pfad, bei Unklarheiten werden die zur Verfügung stehenden Möglichkeiten nach einem zweiten Betätigen angezeigt.

7.3 Die Pfeiltasten ***

Nach dem Kapitel 7 könnte der Eindruck entstehen, dass Linux-Nutzer faule Menschen sind. Nicht nur die Tabulatortaste hilft bei der Eingabe von Befehlen. Mit den Pfeiltasten *Nach-Oben* und *Nach-Unten* kann durch die letzten 500 eingetippten Befehle gescrollt werden um sie durchzusehen und bei Bedarf wiede zu verwenden. Die Befehle sind gespeichert in der Datei *~/.bash_history*. Die ~ gibt an, dass es sich um das *home* Verzeichnis des jeweiligen Nutzers handelt.

7.4 Mittlere Maustaste ***

Unter Linux kann mit *Strg+C* kopiert, mit *Strg+X* ausgeschnitten und mit *Strg+V* eingefügt werden. Zudem kann mit der mittleren Maustaste selektierter Text an eine beliebige Stelle kopiert werden, ohne das der Inhalt der Zwischenablage davon beeinflusst wird.

7.5 Abbruch von Programmen ***

Sollte man ein Programm verlassen, oder dessen Ausführung stoppen wollen, dann gelingt dies mit der Tastenkombination *Strg+Z* oder *Strg+C*.

7.6 Den X-Server neu starten ***

Sollte der X-Server neu gestartet werden müssen, muss nicht sofort das ganze System neu gestartet werden. Mit *Strg+Alt+Backspace* (löschen rückwärts) kann der X-Server neu gestartet werden.

7.7 Linux reagiert nicht mehr ***

Linux *nie* gewaltsam beenden. Das kann äußerst unangenehme Folgen haben. Dass Linux nicht mehr reagiert, sollte so gut wie nie vorkommen. Ein Einloggen über ein Pseudo-Terminal oder über einen anderen Rechner mit ssh gelingt so gut wie immer.

Die Tastenkombination *Strg+Alt+Entf*, oder ein kurzer Druck auf den Einschaltknopf fahren das System sauber für einen Neustart herunter[40].

7.8 Linux ausschalten ***

Nie gewaltsam beenden. Die Vorgehensweise des vorigen Abschnitts sollte (fast) immer funktionieren. Ein gewaltsames Beenden von Linux kann zu Datenverlust bis hin zu einem instabilen System führen. Steht keine grafische Umgebung zur Verfügung kann Linux wie folgt beendet werden:

sudo halt	*(shutdown -h now)*	Ausschalten nur mit root Rechten erlaubt
sudo reboot	*(shutdown -r now[41])*	Neustart nur mit root Rechten erlaubt
sudo reboot	*(shutdown -r +6)*	Neustart in 6 Minuten

40 Dieses Verhalten kann den Wünschen entsprechend angepasst werden.
41 anstelle von *now* kann eine Zeitangabe stehen (*14:30* oder *+10* in 10 Minuten)

7.9 Verzeichnisangaben ***

. (Punkt):	Das aktuelle Verzeichnis
.. (Punkt Punkt):	Das übergeordnete Verzeichnis
~ (Tilde):	Das eigene home Verzeichnis

7.10 Pipes: Ausgaben an einen Befehl weiterleiten ***

Mit so genannten Pipes kann die Ein- Ausgabe eines Befehls gesteuert werden. Mit Hilfe des Pipe-Operators (| *senkrechter Strich*) wird die Ausgabe eines Befehls als Eingabe an einen weiteren Befehl umgeleitet.

Z.B.
more <Datei>: Gibt den Inhalt der Datei stückchenweise aus. Immer 1 Bildschirm nach dem anderen.

ls /etc: Eine mehrere bildschimfüllende Auflistung sämtlicher in /etc vorhandener Verzeichnisse und Dateien wird auf dem Bildschirm ausgegeben.

```
khc_mansearch.pl            zjsdecode
khelpcenter                 zonetab2pot.py
khotkeys                    zsoelim
```

Die vielen Einträge davor sind am user in rasender Geschwindigkeit vorbeigeschossen.

Mit dem Pipe-Operator | können nun beide Befehle miteinander verbunden werden. D.h. Die Ausgabe des einen Befehls (hier ls /etc) wird zur Eingabe des folgenden Befehls (hier more <Datei>)

ls /etc | more:

```
a2p
aconnect
acpi
--More--
```

Es können mehrere Pipes als eine Anweisung zusammengefasst werden.

7.11 Ausgaben in eine Datei umleiten / an eine Datei anhängen ***

Mit dem Operator > wird die Ausgabe eines Befehls in eine Datei umgeleitet. Existiert die Datei nicht, so wird sie erstellt.

Mit dem Operator >> wird die Ausgabe eines Befehls in eine Datei umgeleitet und an das Ende der Datei angehängt. Existiert die Datei nicht, so wird sie erstellt.

```
markus@buero:~$ echo "Welt" > /home/markus/Desktop/nichtda.txt[42]
markus@buero:~$ echo "Hallo" > /home/markus/Desktop/nichtda.txt[43]
markus@buero:~$ echo "Welt" >> /home/markus/Desktop/nichtda.txt[44]
```

42 Die Datei wird erstellt mit dem Inhalt *Welt*
43 Die Datei wird mit dem Inhalt *Hallo* überschrieben

8 Häufige Befehle unter Linux

Die nachfolgenden Befehle werden nicht mit all ihren Möglichkeiten erläutert. Bei Bedarf bitte genauere Informationen über *man* oder das www beziehen. Das Programm *man* und *info* gibt eine Anleitung des angegebenen Befehls auf dem Bildschirm aus. Solche Anleitungen werden *man-pages* genannt. Beendet wird *man* / *info* über *Strg+Z*.

```
klaus@buero:~$ man apropos

APROPOS(1)              Manual Hilfsprogramme              APROPOS(1)

NAME
     apropos - sucht die Manualkurzbeschreibung in der Indexdatenbank

SYNTAX
     apropos [-dhV] [-e|-r|-w] [-m System[,...]] [-M Pfad] Schlüsselwort ...

BESCHREIBUNG
     Innerhalb jeder Manualseite ist eine Kurzbeschreibung vorhanden.  apro-
     pos  sucht  das Schlüsselwort in den Kurzbeschreibungen der Indexdaten-
     bank.  Falls es eine solche nicht im Manualpfad gibt, durchsucht es die
     whatis- Datenban
u.s.w.
```

8.1 Hilfe vom System

!! ***

führt den letzten Befehl nochmals aus.

alias ***

Ein alias-Name kann einem Befehl zugeordnet werden. Gilt nur bis zum nächsten Neustart. Soll ein alias ständig zur Verfügung stehen, so ist er in *~/.bashrc* einzutragen.

```
alias alles='ls -al'
klaus@buero:~$ alles
```

```
klaus@buero:~$ alias doku="echo $1 '#' $2“  >> mitschrift.txt"
klaus@buero:~$ mkdir /probe
klaus@buero:~$ doku !! “Das Verzeichnis Probe wird erstellt“
```

Bei diesem Beispiel wird ein alias *doku* erstellt. Er erwartet 2 Argumente für die Parameter $1 und $2. Alles wird dann an die Datei *mitschrift.txt* angehängt.

44 Das Wort Welt wird angehängt. Die Datei hat nun den Inhalt *Hallo Welt*

apropos <Befehl> ***

apropos <*Stichwort*>. Eine Liste der Befehle, in deren *man-Pages* das Stichwort vorkommt.

```
klaus@buero:~/Desktop$ apropos partition
cfdisk (8)           - Curses based disk partition table manipulator for Linux
fdisk (8)            - Partition table manipulator for Linux
jfs_mkfs (8)         - create a JFS formatted partition
mdrun (8)            - autodetect and activate all raid partitions
mkfs.jfs (8)         - create a JFS formatted partition
parted (8)           - a partition manipulation program
partprobe (8)        - inform the OS of partition table changes
u.s.w
```

apropos -e <Befehl>: sucht nach einer exakten Entsprechung

cal

Zeigt einen Kalender an

```
klaus@buero:~$ cal
   August 2006
So Mo Di Mi Do Fr Sa
      1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

clear ***

Löscht die Konsole.

env ***

Gibt alle Umgebungsvariablen[45] aus.

history

Zeigt die letzten 500 Eingaben des Nutzers in die Konsole an. Es handelt sich um den Inhalt der Datei *~/.bash_history*.

man oder info <Befehl> ***

man <*Befehl*> bzw. i*nfo* <*Befehl*>. Die „Bedienungsanleitung“ für den angegebenen Befehl wird ausgegeben. Mit den Pfeiltasten und mit Bild-Hoch / Bild-Runter kann die Anleitung gelesen werden. Die Anzeige wird mit *Strg+Z* beendet.

45 Kurzreferat

whatis <Befehl>

Eine Kurzinformation über den Befehl wird angezeigt.

```
whatis ls
ls(1)   - list directory contents
```

Hilfe zu Befehlen

Bei sehr vielen Befehlen wird nach Angabe des Parameters – – help eine kurze Hilfe zum Befehl ausgegeben.

```
klaus@buero:~/Desktop$ date --help
Aufruf: date [OPTION]... [+FORMAT]
 oder:  date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]
Zeige die aktuelle Zeit im angegebenen Format an oder setze die Systemzeit.
u.s.w
```

8.2 Informationen Rund um das System

date

Zeigt die aktuelle Uhrzeit mit Datum. *root* kann diese Zeit verändern, welch beim nächsten Neustart allerdings verloren geht. Die Ausgabe kann auf verschiedenste Weise formatiert werden. Z.B. mit %H.%M für die Ausgabe von nur Stunde und Minute. Diese Möglichkeit wird gerne beim Skripten genutzt.

```
klaus@buero:~/Desktop$ date
Mo 21 Aug 22:14:36 CEST 2006
klaus@buero:~/Desktop$ date +%H.%M
22.14
```

Es sei denn root synchronisiert beide mit *hwclock –systohc*.

df ***

Gibt die Festplatten- und Partitionsstruktur mit Auslastung an.

du <Pfad>

Ausgabe des Platzbedarfs einzelner Verzeichnisse mit Unterverzeichnissen. Sinnvoll in Verbindung mit *sort* über eine so genannte Pipe.

```
du /home/klaus
```

```
du /home/klaus | sort
```

Mit der Option -h gestalten sich die Größenangaben lesbarer.

```
du -h /home/klaus | sort
```

Die Verzeichnistiefe kann über die Option –max-depth=*N* gesteuert werden.

```
klaus@buero:~$ du  --max-depth=1 /home/klaus/Desktop/ | sort
148    /home/klaus/Desktop/RadFTP-1.50F
1816   /home/klaus/Desktop/pda
22984  /home/klaus/Desktop/
2744   /home/klaus/Desktop/kurs131
4608   /home/klaus/Desktop/pda2
```

free

Ausgabe der Speicherauslastung des Systems. Bei *free -m* erfolgt die Angabe in Megabyte.

groups ***

Gibt die Gruppenzugehörigkeit des users an

hostname ***

Der Rechnername wird angezeigt

id ***

Der eigene Loginname incl. Gruppenzugehörigkeit wird angegeben. Dieser Befehl kann bei einer Fehleranalyse wichtig sein.

ifconfig ***

Die Konfiguration der Netzwerkschnittstellen wird angezeigt. Bei Übergabe der Option -a wird auf definitiv jede Netzwerkschnittstelle angezeigt.

last

Zeigt an, wann die letzten Logins erfolgten

mount ***

Zeigt an, wie und an welcher Stelle Festplatten bzw. Partitionen eingebunden sind.
Mehr unter *„Verwaltung des Systems“*.

```
klaus@buero:~$ mount
/dev/hdb9 on / type ext3 (rw,errors=remount-ro)
/dev/hda5 on /boot type ext3 (rw)
u.s.w
/dev/hde10 on /media/tausch type vfat (rw,utf8,umask=007,gid=46)
```

pidof <Name> ***

Jedem Programm, jeder Aktivität, jedem laufenden Dienst, ... wird zur Unterscheidung eine eindeutige Prozess-ID zugeordnet. Somit kann: kontrolliert werden ob ein Programm läuft, ihm weniger Rechenleistung[46] zur Verfügung gestellt, oder es beendet werden.

```
klaus@buero:~$ pidof firefox-bin
17817
```

pmount <device> [<label>]

user die Mitglied in der *plugdev* Gruppe sind können USB Geräte einbinden.

pumount

user die Mitglied in der *plugdev* Gruppe sind können USB Geräte aushängen.

ps <-e -a -l>

Zeigt eine Liste aller laufenden Prozesse an[47].

-e: alle
-a: nur die des Terminals
-f: ausführliche Informationen

pstree <-a -p -u>

Alle laufenden Prozesse werden als Baumstruktur ausgegeben.

-a: ausführliche Ausgabe mit Parametern des Programmstarts

-p: die Prozess-ID wird ausgegeben

-u: der Benutzer des Prozesses

```
klaus@buero:~$ pstree -apu

  ├─gnome-terminal,19956,klaus
  │   ├─bash,19958
  │   │   ├─nl,20304
  │   │   ├─pstree,25118 -apu
  │   │   └─sort,21397
  │   ├─bash,24949
  │   │   └─vi,24972 /etc/fstab
```

pgrep <Name> ***

Nun ist es umständlich aus den langen Listen von *ps* bzw. *pstree* die PID bzw. den Namen des gesuchten

46 diese Aufgabe erledigt das Programm renice
47 *ps -ef | grep firefox* liefert Prozess-ID und den Namen des Programms zurück.

Programms zu suchen. Mit *pgrep* kann nach den Prozessen suchen, die <Name> beinhalten.

```
klaus@buero:~$ pgrep firefox
17817
```

pwd ***

Zeigt das aktuelle Arbeitsverzeichnis an.

```
markus@buero:~$ pwd
/home/markus
```

top ***

Eine sich ständig aktualisierende Liste aller laufenden Prozesse wird angezeigt. Die Ausgabe kann mit Strg+Z abgebrochen werden.

uptime

Zeigt an, wie lange das System schon läuft

who

Informationen zu den momentan eingeloggten Benutzer werden angezeigt.

who -b Zeit des letzten Systemstarts
who -r Ausgabe des aktuellen Runlevels

w

Genauere Informationen als bei *who* werden angezeigt.

whoami ***

Anzeige des eigenen Login-Namens

8.3 Umgang mit Dateien/Verzeichnissen

basename <Pfadangabe zu einer Datei>

Gibt den Namen einer Datei ohne Pfadangabe aus.

```
klaus@buero:~$ basename /home/klaus/Desktop/termine.odt
termine.doc
```

bzip2

Siehe packen und entpacken.

cat <Dateiname1> <Dateiname1> usw.

Verbindet mehrere Dateien und gibt das Ergebnis aus. Die Dateien werden dabei nicht verändert.
Die Option -n nummeriert die Zeilen.

cd <Pfadangabe> ***

Zum Wechseln von Verzeichnissen

cd /home/klaus/backup
cd ..
cd ../..

cp <Quelle> <Ziel> ***

Kopieren von Dateien oder Verzeichnissen.

-v: mit ausführlichen Erklärungen
-r: rekursiv, auch enthaltene Verzeichnisse werden kopiert
-i interaktiv: vor einem Überschreiben wird nachgefragt
(-u es wird nur kopiert, wenn die zu kopierende Datei neuer ist.)

cp /home/klaus/backup /home/klaus/alt	nichts passiert, da keine Datei angegeben ist
cp /home/klaus/backup/ /home/klaus/alt*	alle Dateien, keine Unterverzeichnisse werden kopiert
cp -r /home/klaus/backup/ /home/klaus/alt	der Ordner backup wird nach yy kopiert
cp -r /home/klaus/backup/ /home/klaus/alt*	der Inhalt von backup wird nach yy kopiert
cp -riu /home/klaus/backup/ /home/klaus/alt*	nur neuere Dateien werden auf Wunsch kopiert

dirname <Pfadangabe zu einer Datei>

Im Gegensatz zu *basename* wird hier der Pfad zu einer Datei angegeben.

```
klaus@buero:~$ dirname /home/klaus/Desktop/termine.odt
/home/klaus/Desktop
```

file <Dateiname | Verzeichnisname>

Zeigt den Dateityp an.

```
klaus@buero:~/Desktop$ file /etc
/etc: directory
```

```
klaus@buero:~/Desktop$ file /etc/fstab
/etc/fstab: ASCII text

klaus@buero:~/Desktop$ file -i /etc/fstab
/etc/fstab: text/plain; charset=us-ascii
```

-i: Zeigt den „Mime-Typ“.[48] einer Datei an.

gzip

siehe „Packen und Entpacken“

head <Datei> ***

Die ersten Zeilen einer Datei werden ausgegeben.

less <Datei> ***

Der Inhalt der Datei wird angezeigt. Im Gegensatz zu *more* kann man sich mit Hilfe des Cursors innerhalb der Datei bewegen und nach Begriffen suchen[49].

ln <Quelle> <Ziel> ***

Verknüpfungen zu Dateien und Verzeichnissen (symbolische Verknüpfungen) können erstellt werden, so können diese unter unterschiedlichen Namen von unterschiedlichen Orten angesprochen werden.

-i: interaktiv, fragt nach, bevor Dateien überschrieben werden
-s: erstellt eine Symbolische Verknüpfung[50]
-v verbose Mode. Gibt detaillierte Rückmeldungen
-f force; Dateien werden überschrieben.

ln -s /tmp /home/klaus/Desktop/tmp
ln -s /tmp “/home/klaus/Desktop/zeitlich begrenzt“ “ “, wegen des Leerzeichen im Namen.

ln /home/klaus/ferien.odt /home/klaus/Desktop/meins.odt[51]
Beide Dateien *ferien.odt* und *meins.odt* beziehen sich auf dieselbe Datei. Eine Änderung in der einen Datei ändert zeigt sich somit sofort beim öffnen der „anderen“ Datei. Die Dateien können umbenannt werden. Wird eine gelöscht, bleibt die andere weiterhin nutzbar.

ls [Optionen] [Pfad] **

Listet den Inhalt eines Verzeichnisses auf.

48 Was versteht man unter dem Begriff?
49 Eingabe von / <Begriff>
50 auch als Softlink bezeichnet
51 auch als Hardlink bezeichnet

-a: zeigt auch versteckte Dateien an.
-h: Größenangaben menschlich lesbarer (Angaben in MB, kB,)
-l: Zugriffsrechte werden mit ausgegeben.
-s: zeigt die Dateigröße an.
-S: sortiert nach der Dateigröße.
-t: sortiert nach Änderungsdatum
-X: sortiert nach Dateityp

mkdir [-v] [verz1] [verz2] ... ***

Erstellt Verzeichnisse. Auf Wunsch (-v) werden ausführliche Informationen zurückgegeben.

mount [optionen] [-t Dateisystem] [device Quelle] [Zielverzeichnis] ***

Ein Dateisystem (Partitionen, CD ROM; ...) wird in den Verzeichnisbaum *eingebunden*. Root Rechte sind erforderlich. *mount* ist ein sehr umfangreicher Befehl mit einer großen Anzahl von Möglichkeiten (*man mount*). Wird nur *mount* eingegeben, gibt das Programm eine Liste aller derzeit gemounteten Dateisysteme an. In der Datei /etc/fstab[52] ist angegeben, welche Dateisysteme wo beim Systemstart eingebunden werden sollen. Ist eine solche Partition aus irgendeinem Grund nicht gemountet, genügt *mount /Ziel*.

more

Gibt einen Text Seiten weise aus. Es kann nur nach unten gescrollt werden.

mv <Quelle> <Ziel> ***

Zum Verschieben und Umbenennen von Verzeichnissen und Dateien.

-f: erzwingt ein Überschreiben.
-i: fragt bei Dateien vor einem Überschreiben nach.
-v: gibt detaillierte Informationen zurück.

Packen und Entpacken ***

bzip2	
Dateiendung: .tar.bz2	*packt nur einzelne Dateien (Original wird gelöscht)*
auspacken	bunzip2 -v DateiName.bz2
packen	bzip2 DateinName

gz	
Dateiendung: .gz	*packt nur einzelne Dateien (Original wird gelöscht)*
auspacken	gunzip -v DateiName.gz

52 Lass dir den Inhalt der Datei anzeigen.

gz	
packen	gzip DateiName.gz

tar ***	
Dateiendung: .tar	
auspacken	tar -xfv Datei.tar
packen	tar -cfv Ziel.tar verz1 verz2 datei1 ...
ansehen	tar -tvf Datei.tar tar tvf Datei.tar | less
Dateiendung: .tar.bz2	
auspacken	tar -xvjf Datei.tar.bz2
packen	tar -cvjf Ziel.tar.bz2 verz1 verz2 datei1 ...
ansehen	tar -tvjf Datei.tar.bz2 tar tvfj Datei.tar.bz2 | less
Dateiendung: .tar.gz (tgz)	
auspacken	tar -xvzf Datei.tar.gz
packen	tar -cvzf Ziel.tar.gz verz1 verz2 datei1 ...
ansehen	tar -tvzf Datei.tar.gz tar tvfz Datei.tar.gz | less

c = compress
z = komprimieren mit gzip
x = extract (auspacken)
--exclude = einzelne Dateien ausschließen
f = file (Datei)
j = komprimieren mit bzip2
t = list (anzeigen) [evtl. mit less anzeigen lassen]
r = zu einem bestehenden Archiv hinzufügen.

tar -cvzf /data/archiv/Sicherungsdatei.tgz --exclude=.log /home/MyLogin*
tar -cvzf /data/archiv/Sicherungsdatei.tgz --exclude=/home/klaus/yy /home/MyLogin

tar -cvzf /data/archiv/Sicherungsdatei.tgz -X liste.egal /home/MyLogin

So könnte eine Datei (z.B.liste.egal) aussehen

```
/home/klaus
*.log
/home/gdr
```

zip ***	
Dateiendung: .zip	
auspacken	unzip -v DateiName.zip
packen	zip -r DateiName.zip verz1 verz2 datei1 ... zip -rq DateiName.zip verz1 verz2 datei1 ... (q = quick) zip -r Dateiname.zip verz -x *.html (ohne html Dateien) zip -r Dateiname.zip verz -x@liste2.egal (ohne aufgeführte **Dateien**)

So könnte eine Datei (z.B.liste2.egal) aussehen

```
/home/klaus/*   (gilt auch für Unterverzeichnisse)
*.log
/home/gdr/*
```

pwd ***

zeigt das aktuelle Verzeichnis an.

rev

kehrt die Zeichenfolge einer Datei um.

```
klaus@buero:~/depp$ more bsp
aber
hallo
klaus@buero:~/depp$ rev bsp
reba
ollah
```

rmdir ***

Löscht ein leeres Verzeichnis.

rm <Datei1> <Datei2> ... ***

Löscht Dateien oder Verzeichnisse. Standardmäßig werden keine Ordner gelöscht.

-i: Es erfolgt eine Nachfrage bevor gelöscht wird.
-v: Ausführliche Meldungen
-r: rekursiv, Verzeichnisse (auch nicht leere) werden gelöscht.

Der Ordner ~/testen sei ein nicht leerer Ordner, mit mehreren verschachtelten Ordnern und Dateien.

```
klaus@buero:~$ rm -rvi ~/testen
```

Anhand dieses Beispiels kann man sehr schön erkennen, wie der Befehl die Aufgabe rekursiv abarbeitet.

sort ***

Sortiert Ausgabe von Programmen, oder Inhalte von Dateien.

-r: umgekehrte Reihenfolge

sort ~/abc.txt
sort -r ~/abx.txt
ls | sort -r

strings <Datei>

extrahiert alle „lesbaren“ Zeichen aus einer Datei.

tail <Datei> *

Gibt die letzten Zeilen einer Datei aus.

tar

siehe „Packen und Entpacken“

zip

siehe „Packen und Entpacken“

touch *

Leere Dateien werden erstellt. Z.B. die Datei test.txt im eigenen home Verzeichnis.

```
touch ~/text.txt
```

wc <Datei>

Zählt Buchstaben, Zeilen und Wörter einer Datei.

xargs *

find ~ | xargs grep klaus: sucht nach allen Dateien, die den Begriff *klaus* enthalten. Bei *xargs* handelt es sich um einen Befehl, der die mehrzeilige Ausgabe (wie hier von *find*) in einzeilige Ausgaben umwandelt und an den nachfolgenden Befehl weiter gibt. In diesem Beispiel wird die Liste von *find* weitergeleitet an *xargs*, welches die Liste in einzelne Zeilen trennt und an *grep* weiterleitet. grep untersucht die Datei dann auf ihren Inhalt.

8.4 Verwaltung des Systems

<Befehl> &

Prozess im Hintergrund starten.

chmod

Siehe 5.4

chown ***

Der Eigentümer und die Gruppe einer Datei kann geändert werden.
-R: rekursiv -v: verbose

```
 klaus@buero:~$ sudo chown root ./testen
 sudo chown -v root ./testen
Eigentümer von „./testen“ in root geändert
klaus@buero:~$ sudo chown -v root:root ./testen
Eigentümer von „./testen“ in root:root geändert
klaus@buero:~$ sudo chown -v klaus:root ./testen
Eigentümer von „./testen“ in klaus:root geändert
klaus@buero:~$ sudo chown -v klaus: klaus  ./testen
Eigentümer von „./testen“ in klaus: klaus  geändert
```

dd

Kopiert bytegenau Partitionen/Dateisysteme.

e2fsck

Prüft und repariert ein ext2 bzw. ext3 Dateisystem, das nicht eingehängt ist.

kill <PID> ***

Beendet die angegebene Prozess id. kill -9 <PID> bzw. kill -SIGKILL <PID> beendet bedingungslos. Siehe auch ps, pstree, pgrep 8.2.

killall <name> ***

Die Prozesse mit dem übergebenen Namen werden gestoppt, mit killall -9 <name> bedingungslos.

passwd ***

Zum Ändern des Passwortes

umask ***

Mit diesem Tool können die Rechte neu angelegter Dateien und Verzeichnisse voreingestellt werden. Ähnlich zu *chmod* werden die Rechte als Oktalzahlen übergeben, allerdings invertiert. Das bedeutet, dass die mit *umask* gesetzten Rechte bei neuen Dateien *nicht* gesetzt werden. Wird nur umask eingegeben erscheint der aktuell gesetzte Wert.

```
klaus@buero:~$ umask
0022
klaus@buero:~$ umask 0027
klaus@buero:~$ umask
0077
klaus@buero:~$ mkdir eee
```

Zu Beginn lauteten die Rechte: u= rwx- g= rw - o=rw - (umask 0022)
nach der Änderung: u= rwx g= rw - u=- - - (umask 0027)

renice

Bei arbeitsintensiven Aufgaben wie das Konvertieren eines Videos, können solchen Vorgängen weniger Rechenleistung zur Verfügung gestellt werden, um flüssig weiterarbeiten zu können. Root Rechte sind dazu erforderlich. Die Rechenleistung kann in einem Bereich von -20 bis +19 vergeben werden.

```
klaus@buero:~$ sudo renice -20 pidof firefox-bin
oder:
klaus@buero:~$ sudo renice -20 17563
```

su <user> ***

Der Nutzer wird gewechselt. Ohne Angabe des Nutzernames wird zu root gewechselt.

sudo ***

Befehl als root ausführen. Die entsprechende Erlaubnis muss in der Datei /etc/sudoers hinterlegt sein. Diese Datei sollte mit dem Editor visudo bearbeitet werden. Aufruf: *visudo*. Der Aufbau von /etc/sodoers erfolgt nach bestimmten Regeln. Hierzu gibt es genügend Anleitungen im www.

umount <Einhängepunkt>

Darf nur von root ausgeführt werden.

8.5 Netzwerk

ifconfig ***

ifconfig -a: *** Listet alle Netzwerkschnittstellen auf.
ifconfig eth0: *** Informationen über die erste Netzwerkkarte.
*ifconfig eth0 192.168.0.30 netmask 255.255.255.*0 Ip Adresse und Netzmaske zuweisen.

ifconfig eth0:1 192.168.1.30 netmask 255.255.255.0
der ersten Netzwerkkarte eine virtuelle Schnittstelle hinzufügen mit Ip Adresse und Netzmaske.

Diese Einstellungen gehen bei einem Neustart verloren. Permanente Einträge gehören bei Ubuntu in /etc/network/interfaces.

ifdown <Schnittstelle>

Deaktiviert eine Schnittstelle (= ifconfig <Schnittstelle> down)

ifup <Schnittstelle>

Aktiviert eine Schnittstelle (= ifconfig <Schnittstelle> up)

ping <IP | Host> ***

Mit ping <IP | Host> kann die Verbindung zu einem Rechner überprüft werden.

Wird mit Strg+C oder Strg + Z beendet.

8.6 Drucken

lpq ***

Anzeigen von Druckaufträgen

lpr ***

Druckauftrag abgeben. Z.B.lpr datei.txt

lprm ***

Druckaufträge löschen.

lprm <#>: Einzelne Aufträge können anhand der Auftragsnummer aus *lpq* ersichtlich gelöscht werden

8.7 Suchen

find [wo] [Kriterium] [Aktion] ***

Suchen von Dateien ausgehend vom aktuellen Verzeichnis.

-mmin - [n]: Dateien die innerhalb der letzten n Minuten modifiziert wurden.
-mtime - [n]: Dateien die innerhalb der letzten n Tage modifiziert wurden.
-group [name] Dateien der Gruppe
-user [name] Dateien des Nutzers
-name Suche nach dem Dateinamen (Achtung Linux unterscheidet zwischen Groß- und Kleinschreibung)
-iname Verhält sich wie Name. Zwischen Groß- und Kleinschreibung wird nicht unterschieden.
-type [Typ] Dateien der Gruppe
-perm [Recht] Dateien mit den Rechten (als Oktalzahl)

-size +-[n]k Dateien einer größer, kleiner oder gleich (Angabe in KByte)

-fprint [Datei] Das Suchergebnis wird in die angegebene Datei geschrieben

Ein ! vor einer Option negiert diese.
Bei der Option -name kann mit Wildcards gearbeitet werden.

Beispiele:	
find	findet alle Dateien ab dem aktuellen Verzeichnis
find /home/Fritz	findet alle Dateien im angegebenen Verzeichnis
find /tmp -user klaus	findet alle Dattein in /tmp, die Klaus gehören
find / -user klaus -perm 777	findet im ganzen System Dateien von Klaus mit den Rechten 777
find ~ -type d	findet alle Verzeichnisse im Heimatverzeichnis
find ~ type f	sucht alle Dateien im Heimatverzeichnis
find -size +1000k	findet alle Dateien größer als 1 MB (1000 kB)
find -size +1000k -size -1200k	findet alle Dateien zwischen 1 und 1,2 MB
find /tmp -user klaus -atime 3	findet alle Dateien in /opt, die in den letzten 3 Tagen von klaus bearbeitet wurden.
find ~ -name " *.odt"	findet alle OOo Writer Dateien im Heimatverzeichnis
find ~ -name "schule*.odt"	findet alle OOo Writer Dateien, die mit schule beginnen. Eine Datei Schule2006.odt wird nicht gefunden (S ist groß!!)
find ~ -iname "schule*.odt"	findet alle OOo Writer Dateien, die mit schule beginnen. Unabhängig ob groß- oder klein geschrieben. Eine Datei Schule2006.odt wird gefunden.
find ~ ! -name " *.odt"	findet alle Dateien außer OOo Writer Dateien
find ~ -name "schule*.odt" -fprint ~/erg.txt	Schreibt das Suchergebnis in die Datei erg.txt im Heimatverzeichnis

Die Ausgabe einer zu langen Liste kann jederzeit mit Strg+Z bzw. Strg + C abgebrochen werden.

grep <was> <wo> ***

Sucht nach einer Zeichenfolge in einer Datei, oder einer Umleitung durch eine Pipe.

-R: Es wird auch in Unterverzeichnissen gesucht
-i: Groß- und Kleinschreibung wird ignoriert

Beispiele:	
grep "klaus"	findet alle Dateien mit dem Inhalt „klaus" im aktuellen Verzeichnis
grep "klaus" /home/Fritz	findet alle Dateien mit dem Inhalt „klaus" im angegebenen Verzeichnis
grep "klaus" -R /home/Fritz	sucht auch in den Unterverzeichnissen
grep "klaus" -i /home/Fritz	die Groß und Kleinschreibung spielt keine Rolle
grep "klaus" -Ri /home/Fritz/*a*	es wird in Verzeichnissen gesucht, deren aller Dateinamen den String a enthält.

grep können Reguläre Ausdrücke übergeben werden. An dieser Stelle würde das jedoch zu weit führen.

Mit einer Pipe in Kombination mit find kann nach einem String in bestimmten Dateien gesucht werden:

*find ~ iname *test* | xargs*[53] *grep klaus*

whereis ***

Sucht so ziemlich alles, was zu einem Programm gehört. Die Ausführbare-Datei, die man Page und den Quellcode.

which <Befehl | Programm> ***

Gibt die Start-Datei eines Programms aus.

-a: gibt alle möglichen Pfade an.

```
klaus@buero:~/abc$ which firefox
/usr/bin/firefox
klaus@buero:~/abc$ which -a firefox
/usr/bin/firefox
/usr/bin/X11/firefox
klaus@buero:~/abc$
```

53 siehe xargs

9 Anhang

9.1 root, es gibt ihn doch

Wer meint, unbedingt als root unterwegs sein zu müssen, hier folgt die Anleitung:

1. *sudo passwd root* im Terminal eingeben. Nun wird für root ein Passwort festgelegt.
   Das reicht schon um sich als root in einem Terminal einloggen zu können.
2. Einen Login von root in Gnome erlauben. *sudo gedit /etc/gdm/gdm.conf* in der Sektion [security]
   AllowRoot=false in *AllowRoot=true* ändern.
3. root die Administrationsprogramme zur Verfügung stellen.
   sudo gedit /etc/group dort nach der Gruppe *admin* suchen und *root* hinzufügen.

Wer das alles rückgängig machen will kehrt die Schritte 2 + 3 um. Das Passwort für root wird entfernt mit *sudo passwd -l root*. Nun ist alles wieder beim Alten.